IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Bankier, J., et al.	
Serial No.	10/029,638	Group Art Unit: 2152
Filed:	December 19, 2001	Examiner: Troung, Lan Dai T.
Title:	Highly Available Transaction Failure Detection and Recovery For Electronic Commerce Transactions	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

Enclosed is Appellants' Appeal Brief pursuant to 37 C.F.R. § 41.37 in connection with the Notice of Appeal filed June 16, 2008 from the final rejection of Claims 1 – 17 and 19-56 in the Office Action of March 19, 2008 ("Office Action").

A Petition for Extension of Time and a charge authorization for the fees applicable to a Large Entity for this Appeal Brief and the Extension of Time is attached.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, a corporation of the State of Massachusetts. EMC Corporation is a Large Entity.

II. RELATED APPEALS AND INTERFERENCES

There are no pending appeals, interferences or judicial proceedings known to Appellants, to Appellants' legal representative, or to Assignee which may be related to, directly effect, be effected by, or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1 – 17 and 19 - 56 are pending, and Claims 1 – 17 and 19 - 56 stand finally rejected by the Examiner.

IV. STATUS OF AMENDMENTS

There are no un-entered Amendments.

V. SUMMARY OF CLAIMED SUBJECT MATTER

A. Concise Explanation of Subject Matter Claimed

The invention is directed to electronic transaction assurance (eTA) systems, method and program products for processing electronic commerce ("e-commerce") network transactions which comprise messages exchanged between a client and a server. Electronic commerce transactions require highly reliable security to ensure proper completion of the transactions such as sale and purchase, to ensure proper

charges and credits, and to provide confidence to users that their confidential information is not misdirected. The eTA systems, methods and program products of the invention are disposed and used between the client and the server to receive request messages from clients related to the electronic commerce transactions. They detect a failure of the electronic commerce transaction and the actual state of the transaction at failure, select an appropriate recovery action based on the actual state at failure, and provide an expected response messages to the client to mask the failure. The invention further insures that in the event of a failure, recovery and resumption of the electronic transaction occurs from the actual state of the electronic transaction at which the failure occurred.

It is important to assure that e-commerce transactions are correctly and completely processed and that failures are masked from users. Transaction failures adversely impact a user's perception of the security, accuracy and safety of e-commerce transaction processing, and can lead to a user's reluctance to use an on-line e-commerce transaction processing system or outright avoidance of the system.

The invention avoids these problems by preserving the actual state of a transaction at a failure, electing an appropriate action to recover for the failure based upon the actual state, masking the failure from the user by providing an expected response to a transaction request, and resuming the transaction from the state of the transaction at the time of failure.

B. Correspondence Between the Claims and the Specification

The following indicates (in square brackets and bold) the correspondence between the specification pages and lines, drawings and reference characters of the drawings for subject matter defined by the claims on appeal.

1. Independent Claims

a. Claim 1

1. A method of processing electronic commerce transactions **[Fig. 4; pg. 12, ln. 13 – pg. 16, ln. 7]** comprising messages exchanged between a client **[Fig. 1, 104; pg. 7, lns. 4-7]** and a server **[Fig. 1, 106; pg. 7, lns. 10-14]** of a computer network, the method comprising:

establishing a communications connection between the network client and the network server **[pg.7, lns. 6-14; pg. 12, lns. 13-19]** at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receiving a request message from the client **[pg. 12, lns. 12-15]** at the eTA system **[Fig. 4, 410]**, the request message relating to an aspect of the electronic commerce transaction **[pg. 12, lns. 14-15]**;

extracting data from the request message to record a state of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg, 13, ln. 16 - pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**

detecting that a failure has occurred with respect to the electronic commerce transaction **[Fig. 4, 440; pg. 15, lns. 5-8; Fig. 6; pg. 26, lns. 13-14; pg. 28, ln. 16 – pg. 29, ln. 12]**;

determining whether an outcome of the electronic commerce transaction in relation to the request message has failed, and the actual state of the electronic commerce transaction at the failure **[pg. 10. lns. 6 – 13; Fig's 2-3, 212; Fig. 6, pg. 26, lns. 14-15]**;

selecting an appropriate recovery action to recover from the failure based upon said actual state **[pg. 10, lns,. 10-13; pg. pg. 11, lns. 10-15; pg. 15, lns. 12-20; Fig's 2-3, 212, 214]**;

transmitting a response message to the client in accordance with the recovery action, wherein the response message masks the failure from the client by providing an expected response to the request message from the client **[pg. 15, ln. 21 – pg. 16, ln. 3; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**.

b. Claim 14

14. A method of processing electronic commerce transactions **[Fig. 4; pg. 12, ln. 13 – pg. 16, ln. 7]** comprising messages exchanged between a client **[Fig. 1, 104; pg. 7, lns. 4-7]** and a server **[Fig. 1, 106; pg. 7, lns. 10-14]**, the method comprising:

establishing a communications connection between the network client and the network server **[pg. 7, lns. 6-14; pg. 12, lns. 13-19]** at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]** and initiating a series of processes at the eTA system, the processes including:

a transaction monitoring process wherein the eTA system monitors electronic commerce messages that are exchanged between the client and the server in relation to a transaction **[Fig. 4, 435; pg. 14, lns. 5-13]**;

a state capture process wherein the eTA system captures and records information descriptive of one or more states of the transaction **[pg. 9, lns. 4-9**; **Fig's 2-3, 212; Pg. 11, lns. 1-2; pg. 23, lns. 15—16]**

a failure detection process wherein the eTA system determines that a failure has occurred with respect to the transaction and the actual state of the transaction at failure **[Fig's 2-3, 206, 212; pg. 10, lns. 14-17; pg. 28, lns. 16-20]**;

an outcome determination process wherein the eTA system determines the extent to which the server has processed the transaction **[pg. 10. lns. 6 – 13; Fig's 2-3, 212; Fig. 6, pg. 26, lns. 14-15; pg. 29, lns. 13-16]**;

a failure masking process wherein the eTA system masks the occurrence of the failure from the client by sending a response message to the client that is an expected response that the client would have received had the failure not occurred **[pg. 15, ln. 21 – pg. 16, ln. 3; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**; and

a transaction recovery process wherein the eTA system recovers the transaction from the failure based upon said actual state **[pg. 10, lns,. 10-13; pg. pg. 11, lns. 10-15; pg. 15, lns. 12-20; Fig's 2-3, 212, 214]**.

c. Claim 19

19. A method of processing network messages between a network client and a network server **[Fig. 1, 104; pg. 7, lns. 4-7; Fig. 1, 106; pg. 7, lns. 10-14]**, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. lns. 11-14; pg. 12, lns. 13-19; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server **[pg. 12, lns. 12-15; Fig. 4, 410]**;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates **[Fig. 4, 420; pg. 13, lns. 5-7]**;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**;

indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period **[Fig. 4, 440; pg. 14, ln. 17 – pg. 15, ln. 8]**;

determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state to recover from the detected failure **[Fig. 4, 445; pg. 15, lns. 9-20]**;

providing a response message to the network client corresponding to the correct outcome to mask the detected failure **[Fig. 4, 450; pg. 15, ln. 21 – pg. 16, ln. 8; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**; and

logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction **[Fig. 4, 430; pg. 13, ln16- pg. 14, ln. 4; pg. 44, lns. 13-20]**.

d. Claim 33

33. An electronic transaction assurance (eTA) system **[Fig's 1-3, 102, 200; Fig. 9, 900]** that includes:

a communications processor **[Fig. 1, 102; Fig. 3, 200]** that receives electronic commerce transaction messages over a computer network between a customer at a client node **[Fig. 1, 104]** and a server node **[Fig. 1, 106; pg. 7, lns. 1-14; pg. 8, lns. 1-13]**; and

a policy-based policy manager engine **[Fig.'s 2-3, 216]** that manages electronic commerce transaction message processing and resulting customer experience by allowing users of the system to define message processing policies that specify conditions and actions to be taken when any of the specified policy conditions is true to provide transparent failover by masking failures from the customer **[pg. 11, ln. 16 – pg. 12, ln. 3; pg. 31, lns. 1 – 8]**, said masking comprising providing a response message to the customer in accordance with said policies **[pg. 31, lns. 15 -21]**.

e. Claim 37

37. A method for determining the outcome of an electronic commerce transaction **[Fig. 4; pg. 12, ln. 13 – pg. 16, ln. 7]** initiated by a network message between a network client **[Fig. 1, 104; pg. 7, lns. 4-7]** and a network server **[Fig. 1, 106; pg. 7, lns. 10-14]**, the method comprising:

establishing a communications connection between the network client and the network server **[pg.7, lns. 6-14; pg. 12, lns. 13-19]** at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server **[pg. 7, lns. 1-14; pg. 12, lns. 12-15; Fig. 4, 410]**;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates **[Fig. 4, 420; pg. 13, lns. 5-7]**;

generating a transaction identifier associated with the received message and storing the transaction identifier information with the transaction type and message parameters at a back end database **[pg. 30, lns. 3-12, 17-20]**;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**;

resuming the electronic transaction from a failure based upon the preserved state at the failure **[pg. 31, ln. 15 – pg. 32, ln. 6]**; and

masking the failure by providing an expected response to the received message **[pg. 15, ln. 21 – pg. 16, ln. 3; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**.

f. Claim 43

43. A method for measuring the end-to-end response time of each electronic transaction message sent from a network client side to a network server side of a Web site **[pg. 44, ln. 21 – pg. 45, ln. 8]**, the method comprising:

establishing a communications connection between the network client and the Web site network server **[pg.7, lns. 6-14; pg. 12, lns. 13-19]**;

receiving a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates **[Fig. 8; pg. 45, ln. 10 -15]**;

adding code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response message is received by the network client, indicating the end of said electronic commerce transaction **[pg. 44, ln. 21 – pg. 45, ln. 8]**;

generating a transaction identifier associated with each electronic commerce transaction request message received from the network client and storing the transaction identifier information with the transaction type and message parameters at a back end database **[pg. 30, lns. 3-20]**;

preserving a state of the electronic transaction and updating the transaction type and message parameters in response to processing of the electronic transaction **[pg. 27, ln. 4 – pg. 28, ln. 15]**;

resuming the electronic transaction from a failure based upon the preserved state at the failure **[pg. 32, ln. 1 – pg. 33, ln. 9]**; and

masking the failure by providing an expected response to the request message from the network client **[pg. 31, lns. 1-8; pg. 35, lns. 12-16]**.

g. Claim 51

51. A program product **[Fig. 9, 914]** for use in a processor **[Fig. 9, 902]** that executes program steps recorded in a computer-readable media to perform a method of processing network messages between a network client and a network server **[pg. 55, lns. 16-19]**, the program product comprising:

a recordable media **[Fig. 9, 908, 914; pg. 55, lns. 2-9]**;

a program of computer-readable instructions executable by the processor to perform operations **[pg. 55, ln. 16 – pg. 56, ln. 13]** comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. lns. 11-14; pg. 12, lns. 13-19; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server **[pg. 12, lns. 12-15; Fig. 4, 410]**;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates **[Fig. 4, 420; pg. 13, lns. 5-7]**;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**;

indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period **[Fig. 4, 440; pg. 14, ln. 17 – pg. 15, ln. 8]**;

determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state and selecting an appropriate action to recover from the detected failure **[Fig. 4, 445; pg. 15, lns. 9-20]**;

providing an expected response message to the network client to mask the detected failure **[Fig. 4, 450; pg. 15, ln. 21 – pg. 16, ln. 8; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**;

logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction **[Fig. 4, 430; pg. 13, ln16- pg. 14, ln. 4]**.

h. Claim 52

52. A system that processes network messages between a network client and a network server **[Fig's 1-3, 102, 200; Fig. 9, 900]**, the system comprising one or more processors that execute program instructions and receive a data set **[pg. 54, ln. 22 – pg. 56, ln. 13]** , wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. lns. 11-14; pg. 12, lns. 13-19; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receive a network message at the eTA system, which is responsible for the communications between the network client and the network server **[pg. 12, lns. 12-15; Fig. 4, 410]**;

identify a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates **[Fig. 4, 420; pg. 13, lns. 5-7]**;

preserve a state of the electronic commerce transaction and update the transaction type and message parameters in response to processing of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**;

indicate a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period **[Fig. 4, 440; pg. 14, ln. 17 – pg. 15, ln. 8]**;

determine the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state and selecting an appropriate action to recover from the detected failure **[Fig. 4, 445; pg. 15, lns. 9-20]**;

provide an expected response message to the network client to mask the detected failure **[Fig. 4, 450; pg. 15, ln. 21 – pg. 16, ln. 8; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**; and

log and report relevant information about the state and the message parameters of the electronic commerce transaction **[Fig. 4, 430; pg. 13, ln. 16- pg. 14, ln. 4]**.

i. Claim 53

53. A program product **[Fig. 9, 914]** for use in a processor **[Fig. 9, 902]** that executes program steps recorded in a computer-readable media to perform a method for determining the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server **[pg. 55, lns. 16-19]**, the program product comprising:

a recordable media **[Fig. 9, 908, 914; pg. 55, lns. 2-9]**;

a program of computer-readable instructions executable by the processor to perform operations **[pg. 55, ln. 16 – pg. 56, ln. 13]** comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. lns. 11-14; pg. 12, lns. 13-19; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receiving a message related to an electronic commerce transaction at the eTA system **[pg. 12, lns. 12-15; Fig. 4, 410]**;

identifying a transaction type and message parameters included in the received message, thereby defining an electronic commerce transaction to which the message relates **[Fig. 4, 420; pg. 13, lns. 5-7]**;

generating a transaction identifier associated with the received message and storing the transaction identifier information with the transaction type and message parameters at a back end database **[pg. 30, lns. 3-20]**;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**; and

masking a failure of the electronic commerce transaction by providing an expected response message to said received message based upon the preserved state at the failure **[pg. 15, ln. 21 – pg. 16, ln. 3; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**.

j. Claim 54

54. A system that determines the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server **[Fig's 1-3, 102, 200; Fig. 9, 900**, the system comprising one or more processors that execute program instructions and receive a data set **[Fig. 9, 902; pg. 54, ln. 22 – pg. 56, ln. 13]**, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system **[Fig. 1, 102; Fig. 7; pg. lns. 11-14; pg. 12, lns. 13-19; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receive a message related to an electronic commerce transaction at the eTA system **[pg. 12, lns. 12-15; Fig. 4, 410]**;

identify a transaction type and message parameters included in the received ~~network~~ message, thereby defining an electronic commerce transaction to which the message relates **[Fig. 4, 420; pg. 13, lns. 5-7]**;

generate a transaction identifier associated with the received message and storing the transaction identifier information with the transaction type and message parameters at a back end database **[pg. 30, lns. 3-20]**;

preserve a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction **[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14]**; and

masking a failure of the electronic commerce transaction by providing an expected response message to said received message based upon the preserved state at the failure **[pg. 15, ln. 21 – pg. 16, ln. 3; Fig. 6; pg. 31. lns. 1-15; pg. 35, lns. 1-16]**.

k. Claim 55

55. A program product **[Fig. 9, 914]** for use in a processor **[Fig. 9, 902]** that executes program steps recorded in a computer-readable media to perform a method for measuring the end-to-end response time of each electronic commerce transaction

message sent from a network client side to a network server side of a Web site **[pg. 55, lns. 16-19]**, the program product comprising:

a recordable media **[Fig. 9, 908, 914; pg. 55, lns. 2-9]**;

a program of computer-readable instructions executable by the processor to perform operations **[pg. 55, ln. 16 – pg. 56, ln. 13]** comprising:

establishing a communications connection between the network client and the Web site network server **[Fig. 1, 102; Fig. 7; pg. lns. 11-14; pg. 12, lns. 13-19; pg. 25, lns. 17-19; Fig. 6; pg. 26, lns. 12-13]**;

receiving a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates **[Fig. 8; pg. 45, ln. 10 -15]**;

adding code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response message is received by the client, indicating the end of said electronic commerce transaction **[Fig. 8; pg. 45, ln. 10 -15]**;

generating a transaction identifier associated with each electronic commerce transaction request message received from the client and storing the transaction identifier information with the transaction type and message parameters at a back end database **[pg. 30, lns. 3-12, 17-20]**;

preserving a state of the electronic transaction and updating the transaction type and message parameters in response to processing of the electronic transaction

[pg. 10, lns. 3-16; Fig. 4, 430; pg. 13, ln.16 – pg. 14, ln. 4; Fig. 6; pg. 26, lns. 13-14];

resuming the electronic transaction from a failure based upon the preserved state at the failure **[pg. 32, ln. 1 – pg. 33, ln. 9]**; and

masking the failure by providing an expected response to the request message from the network client **[pg. 31, lns. 1-8; pg. 35, lns. 12-16]**.

I. Claim 56

56. A system that measures the end-to-end response time of each electronic commerce transaction message sent from a network client side to a network server side of a Web site **[pg. 44, ln. 21 – pg. 45, ln. 8]**, the system comprising one or more processors that execute program instructions and receive a data set **[Fig. 9, 902; pg. 54, ln. 22 – pg. 56, ln. 13]**, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the Web site network server **[pg.7, lns. 6-14; pg. 12, lns. 13-19]**;

receive a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates **[Fig. 8; pg. 45, ln. 10 -15]**;

add code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response is received by the client,

indicating the end of said electronic commerce transaction **[pg. 44, ln. 21 – pg. 45, ln. 8]**;

generate a transaction identifier associated with each received from the client and store the transaction identifier information with the transaction type and message parameters at a back end database **[pg. 30, lns. 3-20]**;

preserve a state of the electronic transaction and update the transaction type and message parameters in response to processing of the electronic transaction **[pg. 27, ln. 4 – pg. 28, ln. 15]**;

resume the electronic transaction from a failure based upon the preserved state at the failure **[pg. 32, ln. 1 – pg. 33, ln. 9]**; and

masking the failure by providing an expected response to the request message from the network client **[pg. 31, lns. 1-8; pg. 35, lns. 12-16]**.

2. Dependent Claims

2. A method as defined in claim 1, additionally comprising identifying a transaction type associated with the electronic commerce transaction **[Figs. 2-3, 208; pg. 13, lns. 5-7]**.

3. A method as defined in claim 1, wherein each transaction type has an associated transaction model, and additionally comprising maintaining a data base of transaction models and identifying a transaction type by selecting from a transaction model in the database **[Figs. 2-3, 211; pg. 10, lns. 3-10; pg. 27, lns. 18-20]**.

4. A method as defined in claim 3, wherein a transaction model defines an expected response message from the server for a given request message from the

client to thereby enable detecting that a failure has occurred with respect to the transaction when the expected response message is not received **[pg. 10, lns. 3-10]**.

5. A method as defined in claim 3, wherein a transaction model defines suspicious activity such that fraudulent activity is deemed present when the suspicious activity is encountered in a transaction **[pg. 30, ln. 20 – pg. 40, ln. 17]**.

16. A method as defined in claim 14, wherein the failure detection process comprises monitoring for a failure code that is embedded in a response message from the server, wherein the failure code indicates that a failure has occurred **[pg. 14, ln. 19 – pg. 15, ln. 3]**.

20. A method as defined in claim 19, wherein the communications connection is a secure connection **[pg. 47, lns. 12-17]**.

34. An electronic transaction assurance system as defined in claim 33, wherein the policy manager engine masks computer network failures from the customer and generates message interaction with the customer if needed to keep the customer informed of any processing delays and keep the customer engaged in a message dialog to enhance the customer's interaction experience with an e-business Web site at the server node **[pg. 11, ln. 16 – pg. 12, ln. 3]**.

35. A system as defined in claim 34, wherein the eTA system includes multiple eTA nodes **[Fig. 1, pg. 5, lns. 10-12]**, each including a communications processor and a policy manager engine **[Fig's 2-3, 230, 216; pg. 8, ln. 21 – pg. 9, ln. 1; pg. 11, ln. 16 – pg. 12, ln. 3]**.

36. A system as defined in claim 35, wherein at least one of the policy manager engines includes a transaction model that is formulated and built to enable tracking processing of an electronic commerce transaction and storing transaction state, and sharing this state with other eTA nodes in the system **[pg. 9, ln. 16 – pg. 10, ln. 2]**.

38. A method as defined in claim 37, wherein the transaction identifier is stored in an existing field of the back end database **[pg. 30, lns. 17-20]**.

39. A method as defined in claim 37, wherein the transaction identifier is stored in a database table of the back end database **[pg. 30, lns. 17-20]**.

40. A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using an applet executing at the network client **[pg. 30, lns. 13-17]**.

41. A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using an Internet cookie **[pg. 27, lns 10-12; pg. 30, lns. 10-12]**.

42. A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using a browser program at the network client **[pg. 30, lns. 13-15]**.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether Claims 1, 6-10 and 12-15 are unpatentable under 35 U.S.C. §103(a) as obvious over U.S. Application No. 2002/0073211 to Lin et al. ("Lin") in view of U.S. Patent No. 6,381,617 to Frolund et al. ("Frolund") and further in view of U.S. Patent No. 5,958,064 to Judd et al. ("Judd"), as stated on pages 3-10 of the Office Action.

2. Whether Claims 2-5 are unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund and Judd in view of U.S. Patent No. 5,991,750 to Watson et al. ("Watson"), as stated on page 9 of the Office Action.

3. Whether Claims 16 – 17 are unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund and Judd in view of U.S. Patent No. 6,138,159 to Phaal ("Phaal"), as stated on pages 9 - 10 of the Office Action.

4. Whether Claim 11 is unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund and Judd in view of U.S. Application No. 2001/0011235 to Kim et al. ("Kim"), as stated on pages 10 - 11 of the Office Action.

5. Whether Claims 19, 21-32, 51-52 are unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund and Judd in view of U.S. Patent No. 5,991,750 to Watson et al. ("Watson"), as stated on pages 11-16 of the Office Action.

6. Whether Claim 20 is unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund, Judd and Watson in view of U.S. Application No. 2002/0070976 to Tanner et al. ("Tanner"), as stated on pages 16 - 17 of the Office Action.

7. Whether Claims 43-45, 47-50 and 55-56 are unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund and Judd in view of U.S. Patent No. 6,341,285 to Blott et al. ("Blott"), as stated on pages 17-22 of the Office Action.

8. Whether Claims 37-42 and 53-54 are unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Frolund and Judd in view of Watson, as stated on pages 22-25 of the Office Action.

9. Whether Claim 33 is unpatentable under 35 U.S.C. §103(a) as obvious over Lin in view of U.S. Application No. 2002/0087912 to Kashyap ("Kashyap") and further in view of U.S. Patent No. 6,065,017 to Barker ("Barker") and Judd, as stated on pages 25-27 of the Office Action.

10. Whether Claim 34 is unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Kashyap, Barker, Judd and Phaal in view of Wallach, as stated on pages 27-28 of the Office Action.

11. Whether Claims 35-36 are unpatentable under 35 U.S.C. §103(a) as obvious over Lin, Kashyap, Barker, and Judd in view of Phaal, as stated on pages 28-29 of the Office Action.

VII. ARGUMENT

For the reasons which follow, it is respectfully submitted that the grounds of rejection of the claims are improper legally, substantively and factually, are therefore unsustainable and should be reversed.

A. The Standard for Unpatentability Under 35 U.S.C. §103

35 U.S.C. §103 forbids the issuance of a patent only if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time of the invention to one of ordinary skill in the art.

Obviousness under 35 U.S.C. §103 requires that all of the elements of the claims be known in the prior art, that one skilled in the art could have combined the elements as claimed by known methods with no change in their respective functions, and that the combination would have yielded predictable results to produce the claimed invention. (See *KSR v. Teleflex, 550 U.S. ____, 82 USPQ 2d 1385, 1396* (2007))

B. Overview of Argument

The rejections of Claims 1-17 and 19-56 under 35 U.S.C. §103 on various combinations of the nine (9) different references cited by the Examiner are improper and should be reversed for several reasons.

The references do not teach or suggest at least one or more separate limitations of each of the independent Claims 1, 14, 19, 33, 37, 43, and 51-56, and

the combination of the prior art elements in the references according to known methods and their established functions would not have put the claimed invention into the possession of one skilled in the art. Accordingly, the references cannot render the claims obvious under the Section 103 of the statute.

Moreover, the Examiner has failed to establish a *prima facie* case to support the rejections. Rather, the Examiner has improperly applied §103 and committed legal error by failing to consider the claimed invention as a whole, by failing to properly interpret the references and to correctly determine what the references actually taught to one skilled in the art at the time of the invention, and by selectively picking and choosing isolated elements, out of context, from the nine various references (references whose teachings are incompatible and cannot be combined as done by the Examiner) and by using Appellants' disclosure and claims as a roadmap to reconstruct the references in an attempt to fit their non-compatible disclosures together to the reject the claims.

It is respectfully pointed out that the correct application of Section 103 requires a consideration of the invention claimed, as a whole, and what the prior art objectively teaches to one skilled in the art. It is improper to simply collect from disparate references isolated elements and combine the elements using the claims as a roadmap, as was done here in the rejections.

It is respectively submitted, for the reasons that are set out more fully below, that at the time of the invention the claimed invention as a whole would not have been obvious to one of ordinary skill in the art based upon the cited prior art, that the

rejections of the claims on the cited prior art are improper and cannot be sustained, and the rejections should be reversed.

Independent Claims 1, 14, 19, 33, 37, 43, and 51-56 are directed to methods, systems and program products for processing electronic commerce transactions, and call for, in different ways, detecting a failure with respect to an electronic commerce transaction, the actual state of the electronic commerce transaction at the failure, a recovery action from the failure based upon the actual state, and masking the failure by providing the expected response had the failure not occurred to the client.

In particular, for reasons that will be explained below, none of the references teaches or suggests an electronic transaction assurance ("eTA") system or method, as claimed, that is interposed between a client and a server, that receives and transfers electronic commerce transaction messages between the client and server, and that, *inter alia*:

- determines the actual state of an electronic commerce transaction upon detecting a failure of the electronic commerce transaction;
- selects a recovery action based upon that actual state; and
- provides the expected response to a client to mask the failure from the client.

Thus, it is respectfully submitted that none of the cited references, individually or in combination, teaches or suggests these aspects of the claimed invention, and the references cannot render the claimed invention, considered as a whole, obvious and unpatentable.

C. The Rejections Are Improper and Should Be Reversed

1. Claims 1, 6-10 and 12-15 Would Not Have Been Obvious and Unpatentable Under 35 U.S.C. §103(a) Over Lin In View of Frolund and Further In View of Judd

a. Independent Claim 1

Independent Claim 1 is directed to a method of processing electronic commerce transactions comprising request messages and responses exchanged between a client and a server, and the claim includes an electronic transaction assurance (eTA) system and recites, in relevant part:

> detecting that a failure has occurred with respect to the electronic commerce transaction;
>
> determining whether an outcome of the electronic commerce transaction in relation to the request message has failed, and the actual state of the electronic commerce transaction at the failure;
>
> selecting an appropriate recovery action to recover from the failure based upon said actual state;
>
> transmitting a response message to the client in accordance with the recovery action, wherein the response message masks the failure from the client by providing an expected response to the request message from the client.

None of Lin, Frolund or Judd individually or in combination, teaches or suggests a method as set forth in Claim 1, nor would the combination of the references using known methods and according to their known functions produce the claimed invention.

In particular, none of the references teaches or suggests at least an eTA system as claimed or the limitations of Claim 1, and these references would not have put the claimed invention into possession of one skilled in the art.

i. Lin

First, contrary to the Office's statement (Office Action, pg. 3), the claimed eTA does not read on Lin's load balancer. The load balanced of Lin may receive and forward requests between a client and a web server, but it does not perform any of the claimed "extracting", "detecting", "determining" or "selecting" functions recited in the claim. Lin explicitly teaches that the load balancer has a "traffic flow rate measuring module 204 [that] communicates with a browser interface to monitor the data flow rate . . . to distribute a flow of browser requests among the individual webservers 130-134 so as not to overload ant one webserver" (see [0037], lines 8 – 14). Thus, Lin's load balancer merely performs the normal function performed by a load balancer of balancing loads among a plurality of devices. This has nothing to do with the invention of Claim 1, and the presence of a load balance in a client-server network does not suggest an eTA system or its functions as claimed.

Next, Lin teaches a state server separate from the load balancer that monitors and retains session information for recovery of a communications session in the event of a web server failure. Lin teaches that session information may comprises browser and application server data such as an IP address and a "cookie" of a user, ID information about the session, time duration of a session, order of transactions, etc. (see [0028], lines 12-21and [0042], lines 6-14) to enable a failed connection to

be reestablished. Session information does not include the actual state of an electronic commerce transaction, as claimed. Lin merely detects a failed connection between a web browser and a server, not a failed e-commerce transaction, and Lin attempts to reconnect to the webserver to reestablish communications, or else assigns a new webserver to continue a session if the failed web server is shut down. (See Lin [0035]).

Lin is merely concerned with and only teaches recovering from a communications failure by reestablishing the connection. Lin does not disclose or suggest either detecting a failure of an electronic transaction or extracting and recording state information, as claimed. Lin also does not disclose determining the actual state of an electronic commerce transaction at a failure, as recognized by the Examiner (Office Action, pg. 4).

ii. Frolund

The Examiner's citation of Frolund for allegedly disclosing claim limitations missing from Lin improperly attributes teachings to Frolund that are neither disclosed nor suggested by Frolund. Frolund discloses a three-tier transaction processing system that uses a phased commit protocol comprising a series of handshaking messages and responses between client applications, server applications and database systems, as best shown in Figures 2-4. Frolund does not disclose or suggest extracting data from a request message to record a state of an electronic commerce transaction, nor determining whether an outcome of the transaction in relation to a request message has failed, as claimed.

Frolund detects a communications link failure by the failure to receive an expected handshaking protocol message within a timeout (Fig. 4, 82, 88; col. 8, lns. 13-22). Upon detecting a failure, Frolund aborts the transaction by sending a "roll-back" command to the database systems. This terminates the transaction and causes the entire transaction to be retried against a different server. (See col. 6, ln. 61 – col. 7, ln. 10. See also col. 7, lns. 30-37.) Thus, Frolund detects communications session failures, not transaction failures. Moreover, aborting and retransmitting clearly does not mask a failure, and is incompatible with the claimed "masking" of a failure from a client.

Frolund also does not determine either whether the outcome of an electronic commerce transaction in relation to a request message has failed, or the actual state of an electronic commerce transaction at a failure. That Frolund may be capable of performing such functions, as asserted by the Examiner (Office Action, pg. 4), and assuming Frolund is reconstructed in ways not taught or suggested, is not the correct application of 35 U.S.C. §103. It is rather what the reference teaches to those skilled in the art, not whether the reference can be modified to perform a claimed function that is the standard of §103. Frolund's disclosure of using a "heartbeat" or "timeout" to detect a failure halfway in a transaction (Fig. 3, 64; Fig. 4, 82; col. 6, lns 49-54) does not determine the actual state of a transaction, as stated by the Examiner (Office Action, pg. 4). Determining that a failure occurs "halfway" during a database transaction is nothing more than determining a time when a failure occurs, not determining the actual state of the electronic commerce transaction at failure.

Moreover, Frolund also does not select a recovery action based upon the actual state at the failure, as asserted by the Examiner. Rather, Frolund expressly teaches that in the event of a failure of the database transaction before it is completed (by the failure to receive back an "outcome" message), "[t]he STM 18-2 aborts the transaction by sending a 'rollback' 66 command . . . to roll back all transactions for which the outcomes have not been determined" (col. 6, ln 61 – col. 7, ln 1). In the event of failure, an entire transaction is rolled back and retried against another server (col. 7, lns 7-10, 36-37). This is not "selecting an appropriate recovery action to recover from the failure based upon said actual state" of the transaction at failure, as claimed.

Lastly, Frolund does not teach a recovery message that masks the failure from the client by providing an expected response to the client, as claimed. Frolund's "rollback" and "retry" merely abort a transaction and repeat the transaction. This does not comprise an "expected response" that masks the failure from the client, as claimed. Thus, Frolund, like Lin, discloses nothing about selecting a recovery action based upon the actual state of a transaction at a failure, or masking the failure by transmitting a response message to the client that provides an expect response to the client's request message, as claimed. Moreover, as pointed out above, the reference is incompatible with the masking claim limitation, and cannot be combined with the other references, as done by the Examiner.

iii. Judd

Finally, Judd does not teach or suggest detecting and responding to electronic commerce transaction failures. Rather, Judd discloses error recovery from link errors in a transfer of a frame on a link between communications nodes, and recovers errors at the frame level by "retransmitting the last 1 or 2 frames" (col. 7, lns 7-10). Link errors comprise errors such as hardware errors, line faults, ACK time-outs, loss of synchronization, code violations, protocol errors, sequence errors and frame reject errors, for example (see col. 7, ln 16 – col. 8, ln 20). Link errors during frame transmission are not electronic commerce transactions failures, as claimed, but rather communications failures of the same type addressed by Lin.

Further, Judd's teaching of retransmitting frames in the event of link errors to "guarantee completion of data transmission", as stated in the Office Action (pg. 5) does not teach or suggest anything about masking a failure from a client by providing an expected response to the client's request, as claimed. In fact, Judd expressly teaches that upon detecting an error on a link between two nodes in data transfer, one node enters an error recovery mode, discards all data frames and accepts error recovery frames which a master node transfers to all nodes capable of initiating information transfer (col. 2, lns 32-40). During error recovery, the "Link ERP in each node builds a LINK STATUS BYTE and sends it to the other node in the address field of a LINK RESET FRAME" (col. 9, lns 7-9). Retransmitting an "expected" data frame to replace a previous data frame in error clearly does not mask a failure by providing an expected response, as asserted by the Examiner.

From the foregoing, it is clear that Lin, Frolund and Judd have nothing to do with processing electronic commerce transactions that masks failures, and that they

do not individually or in combination teach or suggest the invention of Claim 1 or the claims dependent thereon. Moreover, no combination of the elements of these references according to know methods and in accordance with their respective functions would have predictably yielded the claimed invention. The Examiner has clearly misconstrued or ignored the teachings of the references, and has selectively picked, chosen and reconstructed elements of the references using hindsight afforded by applicants' specification in an attempt to make out a rejection, which is clearly improper. The references would not have placed the claimed invention in possession of one skilled in the art, and they cannot render the claimed invention obvious.

b. Independent Claim 14

Independent Claim 14 is also directed to a method of processing electronic commerce transactions and is somewhat similar to Claim 1. Like Claim 1, Claim 14 calls for establishing a communications connection between a network client and a server at an eTA, but it also explicitly recites that the eTA performs a series of processes that comprise similar to those recited in Claim 1.

Thus, for the reasons discussed above, the references to Lin, Frolund and Judd do not teach or suggest the method of Claim 14, much less performing the method at an eTA between a network client and a server.

Lin's load balancer merely performs the normal load balancing function monitoring data flow requests and distributing requests among web servers to balance loads. The load balancer of Lin does not perform a transaction monitoring

process, or a state capture process, as asserted by the Examiner (Office Action, pps. 6-7), and there is no basis for such an assertion.

Thus, the rejection of Claims 1 and 14 and the claims dependent thereon should be reversed.

Moreover, it is submitted that the Office has not satisfied its obligation to set forth a prima facie case for the rejection of independent Claims 1 and 14 by citing references that clearly do not teach or suggest what they are cited for, and basing the rejection on hindsight reconstruction of the references. Accordingly, the rejections are flawed and should be reversed for this reason also.

c. Claims 6, 8-10

Claims 6 and 8-10 depend from Claim 1 and distinguish over the references for the same reasons that Claim 1 distinguishes.

2. Claims 2-5 Would Not Have Been Obvious over Lin, Frolund and Judd in view of Watson

Claims 2-5 depend from Claim 1 and distinguish over the references for the same reasons Claim 1 distinguishes. Furthermore, Claim 2 calls for identifying a transaction type associated with the electronic commerce transaction, Claim 3 calls for maintaining a database of transaction models and identifying a transaction type by selecting from a model in the database; Claim 4 calls for the model to define an expected response message to enable detection of a failure; and Claim 5 recites that the model defines suspicious activity indicating fraudulent activity. None of the references disclose or suggest these limitations.

Contrary to the Examiner's assertion, Watson deals with handling preauthorized credit transactions using transaction identifiers to identify particular preauthorized transactions. The identifier does not identify a type of electronic commerce transaction, as claimed in Claim 2, but rather one particular transaction.

As to Claim 3, the Examiner has presented no basis for the rejection at all, and has accordingly not satisfied the Office's obligation of setting forth a *prima facie* basis for the rejection. Thus, the rejection is flawed. Moreover, Watson does not disclose either a database of transaction models, or identifying a transaction type by reference to a model, as claimed. Thus, Watson is inapplicable.

3. Claims 16 – 17 Would Not Have Been Obvious Over Lin, Frolund and Judd in View of Phaal

Claims 16 – 17 depend from Claim 14 and distinguish over the references for the same reasons Claim 14 distinguishes.

Furthermore, Claim 16 calls for monitoring for a failure code that indicates a failure which is embedded in a response message from the server. The Examiner has misinterpreted Phaal. The reference discloses nothing about failure codes in a response message from a server. Rather, Phaal discloses redirection of client requests from a failed host server, due to the server going offline, to a new server. Thus, Phaal responds to communications failures, not to transaction failures.

4. Claim 11 Would Not Have Been Obvious Over Lin, Frolund and Judd in View of Kim

Claim 11 depends from Claim 1, and distinguishes over the references for the same reasons Claim 1 distinguishes.

5. Claims 19, 21-32, 51-52 Would Not Have Been Obvious Over Lin, Frolund and Judd in View of Watson

a. Independent Claim 19

Independent Claim 19 includes similar "establishing", "receiving", "identifying" and "preserving" limitations to Claims 1 and 14, and distinguishes over Lin, Frolund, Judd and Watson for the same reasons Claims 1 and 14 distinguish. Furthermore, Claim 19 includes additional limitations that are not disclosed or suggested by these references. These additional limitations include:

> indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;
>
> determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state to recover from the detected failure;
>
> . . . . , and
>
> logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction. (emphasis added)

Again, it is pointed out that the Examiner is attempting to combine incompatible teachings of the references by selectively picking and choosing elements out of context and ignoring what the references actually teach, and is

erroneously apply the standard of §103 by combining the selected elements using hindsight reconstruction.

Frolund's disclosure, for instance, of detecting a communications failure "half-way in a transaction" by the failure to receive an anticipated protocol handshaking response does not indicate a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers, as claimed. Frolund detects communications failures between a client and a server. Moreover, aborting and retrying a transaction upon detecting a failure, as taught by Frolund, clearly cannot mask the failure, as claimed, and no combination of Frolund with the other cited references teaches or produces this claim limitation.

Similarly, Judd's retransmission of a frame of data to replace an erroneous frame does not provide a response message corresponding to the correct outcome to mask the detected failure, as claimed. Judd's teachings are incompatible with this claim limitation, and no combination of Judd with the other cited references teaches or would produce this limitation.

These are examples of the Office ignoring the actual teachings of references, and using Applicant's claims as a road map to pick and combine isolated elements from the references, out of context and without regard to the actual teachings of the references, in an attempt to formulate a rejection under §103. This is an improper application of the standards for determining obviousness.

As to Watson's associating an identifier with a preauthorized transaction, this has nothing to do with identifying a transaction type and message parameters in a received message to define an electronic commerce transaction, for reasons pointed out above.

Finally, Lin's maintaining information about a session is not the same as, and does not teach or suggest, the claimed logging and reporting information about the state of a transaction at failure and the message parameters, as claimed. As pointed out above, session information comprises information about the communications connection between a client and a server, not state information about an electronic commerce transaction.

None of the cited references discloses anything with respect to the determining the correct outcome of an electronic commerce transaction, or providing a corresponding response to mask a detected failure, or logging and reporting, as claimed. The Office cannot establish a *prima facie* rejection of a claim without demonstrating that the prior art renders the claim as a whole obvious. The Office cannot do this by ignoring claim limitations, and selectively picking, choosing, and combining isolated elements from incompatible references, as it has done here.

Thus, it is submitted that the cited references in combination do not and cannot put the invention of Claim 19 or Claims 21 – 32 dependent thereon in the possession of one skilled in the art, or render the claims obvious.

b. Independent Claims 51-52

Independent Claims 51 and 52 comprise a computer product and a system claim that are similar to Claim 19, and distinguish over the cited references for at least the foregoing reasons that Claim 19 distinguishes.

Accordingly, this rejection of Claims 19, 21-32 and 51-52 should be reversed.

6. Claim 20 Would Not Have Been Obvious Over Lin, Frolund, Judd and Watson in View of Tanner

Claim 20 depends from Claim 19 and distinguishes over the references for at least the same reasons Claim 19 distinguishes. Claim 20, moreover, recites that the communications connection between a network client, a network server and the eTA system is a secure connection.

Contrary to the Examiner's rejection, Tanner does not disclose a secure connection. Rather, Tanner discloses a "privacy card" that obscures the identity of the user, which is known to a transaction processing clearing house. Paragraph [0051], referred to by the Examiner discloses the use of a "dummy charge ID" for transactional information to preserve privacy, which is far different from a secure communications channel, as claimed. Tanner's teaching of dummy account information does not teach or suggest a secure communications channel, as set forth in Claim 20. Thus, this rejection of Claim 20 should be reversed.

7. Claims 43-45, 47-50 and 55-56 Would Not Have Been Obvious Over Lin, Frolund and Judd in View of Blott

a. Independent Claim 43

The Examiner's stated reasons for the rejection of independent Claims 43 and 55-56 on Lin, Frolund and Judd are the same as those for rejecting Claims 1, 14 and

19, and are wrong for the same reasons discussed above with respect to those claims. Furthermore, Claim 43 additionally recites:

> adding code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response message is received by the network client, indicating the end of said electronic commerce transaction

Blott's teaching of the use of timestamps in a database system accessed by multiple threads which are stored with data by the database management system for insuring committed data is current (see col. 3, lns. 8-33) does not teach or suggest adding code to a Web page that is served to a client that indicates the start and end of a transaction, as claimed. Blott is not relevant to the claims simply because the reference discloses timestamps. This ignores the context of the teachings of the reference. Database management is concerned with different issues from processing electronic commerce transactions, and Blott teaches nothing about adding code to a Web page to indicate the start and end of a transaction, as claimed.

b. Independent Claims 55-56

Independent Claims 55 and 56 distinguish over the cited art for the same reasons as Claim 43 distinguishes, as do dependent Claims 44-45 and 47-50.

8. Claims 37-42 and 53-54 Would Not Have Been Obvious Over Lin, Frolund and Judd in View of Watson

a. Independent Claims 37 and 53-54

Independent Claim 37 includes recitations similar to those in Claims 19 and 43; and independent Claims 53 and 54 also have recitations similar to Claims 19 and 43. The stated reasons for the rejections of Claims 37 and 53-54 are the same as those previously advanced by the Examiner for Claims 19 and 43. Claims 37 and 53-54 distinguish over Lin, Frolund, Judd and Watson for at least the same reasons Claims 19 and 43 distinguish, and the rejections are improper for the reasons stated above.

b. Dependent Claims 38-42

Dependent Claims 38-42 distinguish over the references for at least the reasons Claim 37 distinguishes, and the rejections of these claims are also improper and should be reversed.

As to the Examiner's rejection of Claims 39-42 by reference to Watson's Figure 5 (Office Action, pp. 24-25), none of Figure 5, Watson's description (col. 12, lns. 16-47) of Figure 5, or elsewhere in the specification does Watson disclose or suggest storing a transaction identifier in a field of a back end database (Claim 38), or in a database table of a back end database (Claim 39), or inserting information in a back end database using an applet (Claim 40), an Internet cookie (Claim 41), or a browser program (Claim 42). Watson's disclosure of an identifier that merely identifies a particular preauthorized transaction is not the same as a server assigning an identifier to a received transaction, as claimed. Merely pointing to a "transaction identifier" in Watson (particularly one that has a different purpose from that claimed) while ignoring the other recitations of the claims fails to consider the invention as a

whole and is insufficient to satisfy the Examiner's obligation to set forth a *prima facie* basis for unpatentability.

Accordingly, the rejection of Claims 37-42 and 53-54 are improper and should be reversed.

9. Claim 33 Would Not Have Been Obvious over Lin in View of Kashyap and Further in View of Barker and Judd

Independent Claim 33 is directed to an electronic assurance (eTA) system comprising a communications processor, and recites in relevant part:

> a policy-based policy manager engine that manages electronic commerce transaction message processing and resulting customer experience by allowing users of the system to define message processing policies that specify conditions and actions to be taken when any of the specified policy conditions is true to provide transparent failover by masking failures from the customer, said masking comprising providing a response message to the customer in accordance with said policies.

Lin's load balancer merely distributes requests according to traffic flow conditions, as recognized by the Examiner. Lin does not teach or suggest that the load balancer has any functionality analogous to the claimed policy-based manager of Claim 33, or that users may establish conditions and actions to be taken when specified policy conditions are true.

Barker's disclosure of a database error recovery system that may be accessed by an administrator (col. 7, lns. 60 -67) and used to repair database errors (col. 16,

lns. 2-16) does not in any way teach or suggest a policy based engine that allows users to define message processing policies and that specify actions to be taken to provide transparent failover, as claimed.

Kashyup is similar to Lin in disclosing a fail-over approach for TCP connections in a peer-to-peer network, and that if a first system running an application crashes, a second peer system assumes the first connection and continues with the application from the point of failure (paragraph [0008]). Thus, Kashyup, like Lin, relates to recovery of a failed connection, not failure of an e-commerce transaction. Kashyap does not disclose or suggest a transparent fail-over and transmitting a response message to a client to providing an expected response and mask a failure of the transaction, as claimed. The mere disclosure in Kashyap of a fail-over policy in no way suggests transparent fail-over, as recited, and Judd, for the reasons previously discussed above, cannot cure this deficiency by his disclosure of aborting and retransmitting a frame when an error occurs. This does not provide transparent failover.

This rejection is another clear example of picking and choosing elements from unrelated prior art disclosures, out of context, and piecing them together to formulate a rejection. This is an improper application of §103.

It is also pointed out that the Examiner's stated motivation to combine the references "to guarantee complete transaction" has nothing to do with providing transparent failover, as recited in the claim, and the rejection fails to provide the required motivation to combine the references to meet the claim.

Thus, the rejection should be reversed.

10. Claim 34 Would Not Have Been Obvious over Lin, Kashyap, Barker, Judd in view of Phaal

Claim 34 depends from Claim 33 and distinguishes over Lin, Kasyap, Barker and Judd for the same reasons Claim 33 distinguishes. Phaal's disclosure of redirecting communications from an assigned server to a new server upon a failure has nothing to do with generating message interaction with a customer to keep the customer informed of delays, as recited in Claim 34, and combining Phaal with the cited references would not produce the claimed invention. Thus, this rejection should be reversed.

11. Claims 35-36 Would Not Have Been Obvious over Lin, Kashyap, Barker, Judd, and Phaal in View of Wallach

Claims 35-36 depend from Claim 34 and distinguish over Lin, Kashyap, Barker, Judd and Phaal for the same reasons. Wallach discloses a fault tolerant database system having copies (88A, 88B and 88C) of a replicated network database directory (col. 4, lns. 25-28). This does not teach or suggest an eTA system having multiple nodes, each including a communications processor and policy manager engine, as recited in Claim 35, or a transaction model that enables tracking, storing and sharing state of an electronic commerce transaction, as recited in Claim 36. Thus, combining Wallach with Lin, Kashyap, Barker, Judd and Phaal would not produce these claims, and this rejection should be reversed.

III. CONCLUSION

In view of the foregoing, it is respectfully submitted that the various rejections of Claims 1 – 17 and 19 - 56 as unpatentable under 35 U.S.C. §103(a) over the cited prior art references are improper, unsustainable, and should be reversed.

Dated: October 14 , 2008

Respectfully Submitted,

/Barry N. Young/

Barry N. Young
Attorney for Assignee/Appellants
Reg. No. 27,744

Customer No. 80280
Law Offices of Barry N. Young
200 Page Mill Road, Suite 102
Palo Alto, CA 94306-2061
Phone: (650) 326-2701
Fax: (650) 326-2799

A. CLAIMS APPENDIX

1. A method of processing electronic commerce transactions comprising messages exchanged between a client and a server of a computer network, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a request message from the client at the eTA system, the request message relating to an aspect of the electronic commerce transaction;

extracting data from the request message to record a state of the electronic commerce transaction;

detecting that a failure has occurred with respect to the electronic commerce transaction;

determining whether an outcome of the electronic commerce transaction in relation to the request message has failed, and the actual state of the electronic commerce transaction at the failure;

selecting an appropriate recovery action to recover from the failure based upon said actual state;

transmitting a response message to the client in accordance with the recovery action, wherein the response message masks the failure from the client by providing an expected response to the request message from the client.

2. A method as defined in claim 1, additionally comprising identifying a transaction type associated with the electronic commerce transaction.

3. A method as defined in claim 1, wherein each transaction type has an associated transaction model, and additionally comprising maintaining a data base of transaction models and identifying a transaction type by selecting from a transaction model in the database.

4. A method as defined in claim 3, wherein a transaction model defines an expected response message from the server for a given request message from the client to thereby enable detecting that a failure has occurred with respect to the transaction when the expected response message is not received.

5. A method as defined in claim 3, wherein a transaction model defines suspicious activity such that fraudulent activity is deemed present when the suspicious activity is encountered in a transaction.

6. A method as defined in claim 1, wherein it is deemed that a failure has occurred with respect to the transaction when a response message is not received from the server in response to the request message.

7. A method as defined in claim 1, wherein a failure has occurred with respect to the transaction when an error code is contained within a response message from the server.

8. A method as defined in claim 1, additionally comprising discarding data that relates to a transaction state that is stored at the server.

9. A method as defined in claim 1, wherein determining whether an outcome of the transaction in relation to the request message has succeeded or failed comprises sending a query message to the server to inquire as to the state of the transaction.

10. A method as defined in claim 1, wherein the appropriate recovery action comprises re-directing the request message to another server in order to complete the failed transaction.

11. A method as defined in claim 1, wherein the commerce electronic transaction relates to adding an item to a shopping cart, and wherein extracting data from the request message to record a state of the electronic transaction comprises recording the contents of the shopping cart using data contained in the request message.

12. A method as defined in claim 1, wherein the response message masks the failure from the client such that the client is oblivious to the failure.

13. A method as defined in claim 1, wherein the response message masks the failure from the client such that the client is compensated for the failure.

14. A method of processing electronic commerce transactions comprising messages exchanged between a client and a server of a computer network, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system and initiating a series of processes at the eTA system, the processes including:

a transaction monitoring process wherein the eTA system monitors electronic commerce messages that are exchanged between the client and the server in relation to a transaction;

a state capture process wherein the eTA system captures and records information descriptive of one or more states of the transaction;

a failure detection process wherein the eTA system determines that a failure has occurred with respect to the transaction and the actual state of the transaction at failure;

an outcome determination process wherein the eTA system determines the extent to which the server has processed the transaction;

a failure masking process wherein the eTA system masks the occurrence of the failure from the client by sending a response message to the client that is an expected response that the client would have received had the failure not occurred; and

a transaction recovery process wherein the eTA system recovers the transaction from the failure based upon said actual state.

15. A method as defined in claim 14, wherein the state capture process comprises capturing packets contained in electronic request messages from the client to the server and storing the packets with an identifier associated with a particular transaction between the client and the server.

16. A method as defined in claim 14, wherein the failure detection process comprises monitoring for a failure code that is embedded in a response message from the server, wherein the failure code indicates that a failure has occurred.

17. A method as defined in claim 14, wherein the failure detection process comprises monitoring for a response message from the server and deeming that a failure has occurred if a response message is not received within a predetermined time span.

18. (Cancelled)

19. A method of processing network messages between a network client and a network server, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction;

indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;

determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state to recover from the detected failure;

providing a response message to the network client corresponding to the correct outcome to mask the detected failure; and

logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction.

20. A method as defined in claim 19, wherein the communications connection is a secure connection.

21. A method as defined in claim 19, wherein indicating a detected failure comprises monitoring operation of hardware and software components of the communication connection.

22. A method as defined in claim 21, wherein monitoring comprises intercepting responses from the back-end servers and inspecting the enclosed messages to check for failures and formulating an appropriate response and sending it to the network client.

23. A method as defined in claim 19, wherein the network messages are transmitted in accordance with Internet protocol processing.

24. A method as defined in claim 19, further including training the transaction assurance system to classify and identify transaction types using a supervised machine learning technique, thereby enabling the system to be deployed in different e-business environments with different transaction models.

25. A method as defined in claim 24, wherein a transaction model is associated with a type of electronic commerce transaction such that the transaction model defines expected network activity with respect to the associated type of electronic commerce transaction.

26. A method as defined in claim 25, wherein the expected network activity comprises response messages that are expected from the server in response to request messages from the client.

27. A method as defined in claim 25, additionally comprising detecting ~~for~~ a failure in a network backend system by comparing a response message from the backend system to an expected response message defined in a transaction model.

28. A method as defined in claim 25, wherein determining the correct outcome of the transaction is accomplished by determining an expected outcome that is defined in a transaction model.

29. A method as defined in claim 25, providing a response message to the network client with an appropriate message corresponding to the expected outcome to mask the detected failure is accomplished by using a response message that is defined in a transaction model.

30. A method as defined in claim 25, wherein a transaction model defines suspicious activity and additionally comprising determining that fraudulent activity is present when the suspicious activity is encountered in a transaction.

31. A method as defined in claim 25, wherein a transaction model defines a billing charge for a type of transaction and additionally comprising tabulating billing charges based on the number of times that an actual transaction defined in a transaction model is encountered.

32. A method as defined in claim 19, wherein the system permits resumption of communication with wireless clients when the wireless clients reconnect to the system, without having to resubmit requests they made before disconnecting from the system due to losing wireless signal.

33. An electronic transaction assurance (eTA) system that includes:

a communications processor that receives electronic commerce transaction messages over a computer network between a customer at a client node and a server node; and

a policy-based policy manager engine that manages electronic commerce transaction message processing and resulting customer experience by allowing users of the system to define message processing policies that specify conditions

and actions to be taken when any of the specified policy conditions is true to provide transparent failover by masking failures from the customer, said masking comprising providing a response message to the customer in accordance with said policies.

34. An electronic transaction assurance system as defined in claim 33, wherein the policy manager engine masks computer network failures from the customer and generates message interaction with the customer if needed to keep the customer informed of any processing delays and keep the customer engaged in a message dialog to enhance the customer's interaction experience with an e-business Web site at the server node.

35. A system as defined in claim 34, wherein the eTA system includes multiple eTA nodes, each including a communications processor and a policy manager engine.

36. A system as defined in claim 35, wherein at least one of the policy manager engines includes a transaction model that is formulated and built to enable tracking processing of an electronic commerce transaction and storing transaction state, and sharing this state with other eTA nodes in the system.

37. A method for determining the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message related to said electronic commerce transaction at the eTA system, which is responsible for the communications between the network client and the network server;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

generating a transaction identifier associated with the received message and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction;

resuming the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the received message.

38. A method as defined in claim 37, wherein the transaction identifier is stored in an existing field of the back end database.

39. A method as defined in claim 37, wherein the transaction identifier is stored in a database table of the back end database.

40. A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using an applet executing at the network client.

41. A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using an Internet cookie.

42. A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using a browser program at the network client.

43. A method for measuring the end-to-end response time of each electronic transaction message sent from a network client side to a network server side of a Web site, the method comprising:

establishing a communications connection between the network client and the Web site network server;

receiving a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message

parameters, thereby defining an electronic commerce transaction to which the message relates;

adding code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response message is received by the network client, indicating the end of said electronic commerce transaction;

generating a transaction identifier associated with each electronic commerce transaction request message received from the network client and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserving a state of the electronic transaction and updating the transaction type and message parameters in response to processing of the electronic transaction; ~~and~~

resuming the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the request message from the network client.

44. A method as defined in claim 43, wherein the transaction identifier is stored in a database table of the back end database.

45. A method as defined in claim 43, wherein storing the transaction identifier comprises inserting information into the back end server database using an applet executing at the network client.

46. A method as defined in claim 43, wherein storing the transaction identifier comprises inserting information into the back end server database using an Internet cookie.

47. A method as defined in claim 43, wherein storing the transaction identifier comprises inserting information into the back end server database using a browser program at the network client.

48. A method as defined in claim 43, wherein the eTA system includes multiple eTA nodes.

49. A method as defined in claim 48, wherein a received network message is directed to one of the available eTA nodes.

50. A method as defined in claim 43, further including:

communicating information relating to the communications connection at the selected node to one or more of the remaining eTA nodes;

detecting the removal or failure of the selected eTA node from operation during processing of the received network message;

preserving the state of the selected eTA node processing with respect to the received network message in one or more nodes of the eTA system; and

moving the communications connection from the removed selected node to one of the remaining eTA nodes that are still operating in accordance with the preserved node state such that the network client and network server that were using the selected eTA node do not see any interruption in their communications.

51. A program product for use in a processor that executes program steps recorded in a computer-readable media to perform a method of processing network messages between a network client and a network server, the program product comprising:

a recordable media;

a program of computer-readable instructions executable by the processor to perform operations comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction;

indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;

determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state and selecting an appropriate action to recover from the detected failure;

providing an expected response message to the network client to mask the detected failure;

logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction.

52. A system that processes network messages between a network client and a network server, the system comprising one or more processors that execute program instructions and receive a data set, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receive a network message at the eTA system, which is responsible for the communications between the network client and the network server;

identify a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

preserve a state of the electronic commerce transaction and update the transaction type and message parameters in response to processing of the electronic commerce transaction;

indicate a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;

determine the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state and selecting an appropriate action to recover from the detected failure;

provide an expected response message to the network client to mask the detected failure; and

log and report relevant information about the state and the message parameters of the electronic commerce transaction.

53. A program product for use in a processor that executes program steps recorded in a computer-readable media to perform a method for determining the

outcome of an electronic commerce transaction initiated by a network message between a network client and a network server, the program product comprising:

a recordable media;

a program of computer-readable instructions executable by the processor to perform operations comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a message related to an electronic commerce transaction at the eTA system;

identifying a transaction type and message parameters included in the received message, thereby defining an electronic commerce transaction to which the message relates;

generating a transaction identifier associated with the received message and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction; and

masking a failure of the electronic commerce transaction by providing an expected response message to said received message based upon the preserved state at the failure.

54. A system that determines the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server, the system comprising one or more processors that execute program instructions and receive a data set, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receive a message related to an electronic commerce transaction at the eTA system;

identify a transaction type and message parameters included in the received ~~network~~ message, thereby defining an electronic commerce transaction to which the message relates;

generate a transaction identifier associated with the received message and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserve a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction; and

masking a failure of the electronic commerce transaction by providing an expected response message to said received message based upon the preserved state at the failure.

55. A program product for use in a processor that executes program steps recorded in a computer-readable media to perform a method for measuring the end-to-end response time of each electronic commerce transaction message sent from a network client side to a network server side of a Web site, the program product comprising:

a recordable media;

a program of computer-readable instructions executable by the processor to perform operations comprising:

establishing a communications connection between the network client and the Web site network server;

receiving a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates;

adding code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response message is received by the client, indicating the end of said electronic commerce transaction;

generating a transaction identifier associated with each electronic commerce transaction request message received from the client and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserving a state of the electronic transaction and updating the transaction type and message parameters in response to processing of the electronic transaction;

resuming the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the request message from the network client.

56. A system that measures the end-to-end response time of each electronic commerce transaction message sent from a network client side to a network server side of a Web site, the system comprising one or more processors that execute program instructions and receive a data set, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the Web site network server;

receive a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates;

add code to the Web page served to the network client that records the time when a request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response is received by the client, indicating the end of said electronic commerce transaction;

generate a transaction identifier associated with each received from the client and store the transaction identifier information with the transaction type and message parameters at a back end database;

preserve a state of the electronic transaction and update the transaction type and message parameters in response to processing of the electronic transaction;

resume the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the request message from the network client.

B. EVIDENCE APPENDIX

NONE

C. RELATED PROCEEDINGS APPENDIX

NONE